

Base Distribution - Work Package # 47895

Status:	New	Priority:	Could have
Author:	Christian Müller	Category:	
Created:	2013-05-03	Assigned To:	
Updated:	2013-09-30	Due date:	
Subject:	Taxonomy solution		
Description			
Create a generic taxonomy solution to avoid "insular solutions"			
Target Audience: developers that use taxonomies (tags, categories) in their plugins / websites			
Responsible: Christian Müller <christian (dot) mueller (at) typo3 (dot) org>			
Implemented by:			
Amount:			
Version:			
Planned Implementation Timeframe:			
Motivation			
TYPO3 Neos should provide a generic taxonomy solution to serve websites that are more based on a category model instead of pages and furthermore to have a common base for tagging and categorizing data in Neos plugins, like news, media assets and so on...			
<ul style="list-style-type: none">- MUST: Taxonomy model (based on either TYPO3CR or domain models)- MUST: import/export service for taxonomies- MUST: integration to neos to use as side wide structuring system (menu based on taxonomies, content based on taxonomies).- MUST: clearly documented workflow on how to use taxonomies in plugins- COULD: interface and neos module to manage taxonomies			

History

#1 - 2013-05-08 10:03 - Soren Malling

As soon as there is a plan for the ui/ux I will love to give this a go

#2 - 2013-05-24 12:53 - Sebastian Kurfuerst

FYI: there is an important discussion on the Forum going on about that -- see <http://forum.typo3.org/index.php/t/196982/>

#3 - 2013-05-24 17:53 - Jacob Floyd

I think Aske explained some of the Taxonomy goals very concisely (Nice!):

My idea is to have a very generic package called TYPO3.Taxonomy that has support for adding categories or tags for nodes and domain models. It should come with a powerful Neos module where the integrator can choose which category/tag groups can be used for which elements. It has to support multiple references per element and also be easy for the developer to use.

Looking into my magic crystal ball (read: this is my brain dump), here's what I think would be cool in the future (think big, and then pick the best parts to

implement first):

- A decent ontology editor¹ as a NEOS Module
 - OWL² and RDF-S³ support - Import, and export OWL files; possibly persist ontologies in owl files (in a site package, or an company-wide package shared across sites) for roundtripping with external tools (like VUE⁴).
 - Good Visual editing of the map of words/tags and their relation to one another (maybe embed VUE -- It's a Java app that could be launched with JWS)
 - Possible uses of ontologies built like this:
 - Build trees of pages based on an ontology
 - Maybe it could recommend to an editor: this page is on dogs, you've got labradors in your ontology/mindmap - do you want to add a section or a new page on labradors?
 - Recommending tags for a page that uses words from the ontology
 - Maybe you could explore the site from the ontology - each word/tag/node in the ontology could have a link to search for pages/nodes that include that word or variations of that word and offer to tag those pages that include that word.
 - Or, it could show the recommended tags to an editor who is currently editing a page and is using words from the ontology.
 - If we could have a spell-check-like feature that suggests people use a different word to match the company's ontology that would be cool.
 - Some organizations have ontologies to tell content writers/editors that they should use word foo instead of word bar to mean definition baz.
 - This could be a boon for translations when large organizations say that a word bar with definition baz should always be translated as word foo in the foreign language.
 - Maybe this could use VIE to markup content that is tagged or could be tagged...
- Of course we also need the typical usage of letting users tag pages/posts/nodes or media assets with whatever tags they like (like drupal and wordpress)
- And we want to be able to generate word clouds of tags used in the site or just a particular section of the site (a plugin?)
- It needs to be easy for developers to reuse ontologies
 - They should be able to let users tag stuff in their domain model (A tagging/taxonomy aspect maybe? Or a special annotation that turns it on for an entity in a domain model) *[like Aske's suggestion]*
 - Something to help in searching for stuff that's tagged
 - they should be able to tie into the ontology editor and add functionality like, say, searching wikipedia, or pulling images from external libraries, or...
 - Packages should be able to have a generic ontology view of their domain (think of it like a list view, only showing a web of their domain) or the content that has been persisted in their domain (so I could see a web of images related to a word in the Media Browser). This could be kind of like VisualThesaurus.com or VUE or FreeMind or other mind mapping tools - it's a mind map of content. So, Taxonomy would provide the skeleton/boilerplate view (fluid templates), and then the package, like Media Browser, would add the content specific functionality around it.
 - The ontologies could be used in PackageBuilder to kickstart packages that use portions of an ontology as the domain model. Then, when the ontology gets changed, the developer could have a feature to help him update the code with the new domain vocabulary defined in the ontology (/me thinks that could be a scary but wicked cool feature)
 - Of course, tags should have many-to-many relations where tags can be related to other tags, and content can be related to many tags *[like Aske's suggestion]*
- And the feature where an integrator can see the ontology and map tags (or webs/hierarchies of tags) to where they can be mapped (this can only be mapped to whole pages/posts, and this can only be mapped to individual content elements...) Or, the integrator can have a list of levels and choose the tags that can be mapped onto it (two different views of the information depending on who is doing it and what their focus is - the ontology or the content nodes) *[like Aske's suggestion]*

[1] http://en.wikipedia.org/wiki/Ontology_editor

[2] http://en.wikipedia.org/wiki/Web_Ontology_Language

[3] http://en.wikipedia.org/wiki/RDF_Schema

[4] <http://vue.tufts.edu/>

Visual Understaing Environment (VUE) supports OWL and RDF-S (see <http://vue.tufts.edu/features/index.cfm>)

#4 - 2013-09-25 11:36 - Soren Malling

Wow, Mr. Floyd blew my mind with all his ideas :-)

I'm bringing stuff down to a more low-tech level to get some more background technical stuff started.

My goal is to create a Taxonomy package that can be used by Flow apps in general, including Neos.

So here, goes some "model talk":

The two models I'm currently working with is a **Vocabulary** (a container, could be site specific for example in Neos context) and a **Term** which exists inside a Vocabulary

Working with terms in Neos content is done through a inspector element like the ImageVariant giving you the UI to choose your Terms.

On top of this, I see a possibility of implementing SPARQL and semantic annotation as separate services

One important note: This should work in terms of the Neos workspace as well, so added taxonomy to a element should not be published before the NodeData object is published

#5 - 2013-09-30 12:35 - Soren Malling

My first code commits of the taxonomy work

The taxonomy Flow Application that can be used by every other Flow app

<https://github.com/sorenmallig/TYPO3.Taxonomy>

The Neos module, and implemented with nodes

<https://github.com/sorenmallig/TYPO3.Neos.Taxonomy>

The implementation in Node is done on TYPO3.Neos:Node level, meaning that every other Node object/entity created by an integrator will automatically have taxonomy attached to there objects.

<https://github.com/sorenmallig/TYPO3.Neos.Taxonomy/blob/master/Configuration/NodeTypes.yaml>

First concept of the Inspector implementation is to have it working with a "auto suggest" with a result like this, for choosen terms

<https://www.evernote.com/shard/s58/sh/858d43db-6d0b-419b-8e83-f39f2d75610f/c7789f6b3f689143ced3fa210955974d>

Future stuff:

Eventually, when you create a new Vocabulary you can select a import source and have terms imported from that souce, given a importer for that format. Setting up a more complex Taxonomy grouping (like Jacob described) giving better chances to autotag/suggest content